Project Report
VLG-1

Very Large Graphs for Information Extraction (VLG)

Summary of First-Year Proof-of-Concept Study

B.A. Miller
N.T. Bliss
N. Arcolano
M.S. Beard
J. Kepner
M.C. Schmidt
E.M. Rutledge

20 August 2013

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

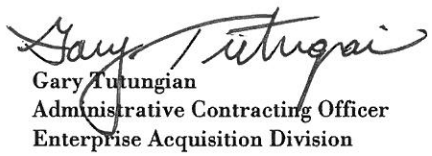
Prepared for the Intelligence Advanced Research Projects Activity (IARPA) under
Air Force Contract FA8721-05-C-0002.

This report is based on studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This work was sponsored by the Intelligence Advanced Research Projects Activity (IARPA) under Air Force Contract FA8721-05-C-0002.

The IARPA PAO has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER



Gary Tutungian
Administrative Contracting Officer
Enterprise Acquisition Division

Massachusetts Institute of Technology
Lincoln Laboratory

Very Large Graphs for Information Extraction (VLG)
Summary of First-Year Proof-of-Concept Study

B.A. Miller
N. Arcolano
M.S. Beard
J. Kepner
M.C. Schmidt
Group 53

E.M. Rutledge
Group 102

N.T. Bliss
formerly Group 110

Project Report VLG-1

20 August 2013

Lexington Massachusetts

This page intentionally left blank.

EXECUTIVE SUMMARY

In numerous application domains relevant to the Department of Defense (DoD) and the Intelligence Community (IC), data of interest take the form of entities and the relationships between them, and these data are commonly represented as graphs. In its role as a DoD-sponsored federally funded research and development center (FFRDC), MIT Lincoln Laboratory (MIT LL) assisted the Intelligence Advanced Research Projects Activity (IARPA) with independent scientific research and analysis of uncued detection techniques for anomalous characteristics within massive graphs whose structure and content change over time.

Under the Very Large Graphs for Information Extraction (VLG) effort—a one-year proof-of-concept study—MIT LL developed novel techniques for anomalous subgraph detection, building on tools in the signal processing research literature. These techniques have the potential to bring powerful new capabilities to the analytic workforce. This report documents the technical results of this effort, many of which have been published in peer reviewed venues [1–6]. Under this effort, two datasets—a snapshot of Thompson Reuters' Web of Science database and a stream of web proxy logs—were parsed, and graphs were constructed from the raw data. From the phenomena in these datasets, several algorithms were developed to model the dynamic graph behavior, including a preferential attachment mechanism with memory (where the probability of new attachment is given by a linear combination of recent attachment rates), a streaming filter to model a graph as a weighted average of its past connections, and a generalized linear model for graphs where connection probabilities are determined by additional side information or metadata. A set of metrics was also constructed to facilitate comparison of techniques.

The study culminated in a demonstration of the algorithms on the datasets of interest, in addition to simulated data. Performance in terms of detection, estimation, and computational burden was measured according to the metrics. Among the highlights of this demonstration were the detection of emerging coauthor clusters in the Web of Science data, detection of botnet activity in the web proxy data after 15 minutes (which took 10 days to detect using state-of-the-practice techniques), and demonstration of the core algorithm on a simulated 1-billion-vertex graph using a commodity computing cluster.

This page intentionally left blank.

ACKNOWLEDGMENTS

The authors wish to thank Mr. Robert Bond for supporting this work and reviewing this report. We also thank the LLGrid team, who enabled much of the work documented herein.

This page intentionally left blank.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF FIGURES

LIST OF FIGURES
(Continued)

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION

In numerous applications, a set of relationships, connections, or transactions between entities is considered, with the objective of finding a small number of entities that are engaging in some activity of interest. It may be desirable, for example, to find people in a social network that exhibit unnoticed influence over many other people, or computers in a network that have been infected by malicious software. Regardless of the specific application, the problem is to detect and identify a set of entities that behave in a coordinated fashion that does not typically appear in normal activity. The set of entities and relationships in these problems can be represented as a graph.

Graphs are combinatorial mathematical objects, and have been used for hundreds of years as abstract representations of relationships between entities. Recently, with the advent of new data sources such as the world wide web and online social networks, graphs have become increasingly popular in the representation of massive datasets, sometimes with billions of entities. For datasets of these sizes, many of the traditional combinatorial graph algorithms are intractable in practice.

Building on previous work on anomalous subgraph detection, Lincoln Laboratory conducted a one-year proof-of-concept study to determine the capabilities and challenges in uncued anomaly detection in massive graphs. Under this effort, large graphs were generated from two data sources, and their phenomena were studied to incorporate models for their behavior into the existing framework for subgraph detection. Challenges in data handling, algorithm development, and the development of performance are documented in this report, in addition to the demonstration of the new algorithms developed under this effort on the massive datasets used.

The remainder of this report is organized as follows. The baseline algorithmic framework for anomalous subgraph detection is presented in Section 2. Section 3 provides an overview of the present study. The datasets used, algorithms developed, and metrics chosen are documented in Sections 4, 5, and 6, respectively. Section 7 demonstrates algorithms performance, evaluated by the chosen metrics, on both simulated data and the real, large datasets prepared. Section 8 provides a summary and outlines future work.

This page intentionally left blank.

2. SIGNAL PROCESSING FOR GRAPHS

To detect anomalous coordination in relational data, the entities and relationships are represented as a graph. A graph $G = (V, E)$ is a pair of sets, a set of vertices V denoting entities, and a set of edges E, which represent relationships. We will refer to the problem of detecting a subset of V that is engaged in anomalous behavior as the *subgraph detection* problem.

The focus of a recent Lincoln Laboratory technical effort known as Signal Processing for Graphs (SPG) is to develop a computationally tractable framework to address the subgraph detection problem in large relational datasets. While the use of graphs to represent relationships has become increasingly popular, the techniques for the detection of behavior of interest within the larger set of relationships tend to be either designed for a specific application [7] or use overly simple background models [8]. The purpose of this effort is to design a broadly applicable framework that is agnostic to both the data model and the application space. This will facilitate specialization of subgraph detection algorithms for diverse, specific application requirements, as well as various data sources.

The framework developed under the initial, internally funded SPG effort uses spectral analysis of graph residuals to determine the presence of an anomalous subgraph and locate it within the network. Several algorithms were developed to detect subsets of vertices that stand out from the "normal" background behavior, based on the expected value of the graph's topology. The expected value is generated by fitting the observed data to a given model, and several linear-algebraic algorithms were proposed for detection of anomalous subgraphs. In this early work, the residuals model chosen was the modularity matrix [9], which considers the fit of the graph to a model in which the probability of an edge occurring between two vertices is proportional to the product of their degrees. Three increasingly complex algorithms were developed for subgraph detection. The first considered the graph residuals matrix in its principal two components, and performed a chi-squared test for independence in this space [10]. The complexity of this algorithm is dominated by computation of the eigenvectors, which costs $O(|V|+|E|)$ time. To detect smaller subgraphs, an algorithm was developed that considers the L_1 norm of the top k eigenvectors, and declares the presence of an anomaly if an eigenvector of a certain rank, that has been unit-normalized in an L_2 sense, has an exceptionally small L_1 norm [11]. This allows detection of subgraphs that stand out in a single eigenvector, and costs $O(|E|k + |V|k^2)$ time. A still more complicated algorithm uses sparse principal component analysis (PCA) [12, 13], a statistical technique to find large variance in the space of a few covariates, to find large *residuals* in the space of a few *vertices* [14]. This technique is much more complex, costing $O(|V|^4\sqrt{\log|V|})$ for a constant error tolerance, but is able to find outliers that do not stand out in a single eigenvector. Detection performance examples are shown in Figure 1, demonstrating the ability of more complex algorithms to detect smaller, subtler anomalies than the less expensive ones.

The techniques for anomaly detection in static graphs were extended to work with graphs that have time-varying topologies. The concept of matched filtering is applied in the context of subgraph detection, using the weighted sum of residuals over time to boost signal power [15]. This approach allows the detection of anomalies that are not detectable in a given instance. This is demonstrated in the results shown in Figure 2. Recall from Figure 1 that, using

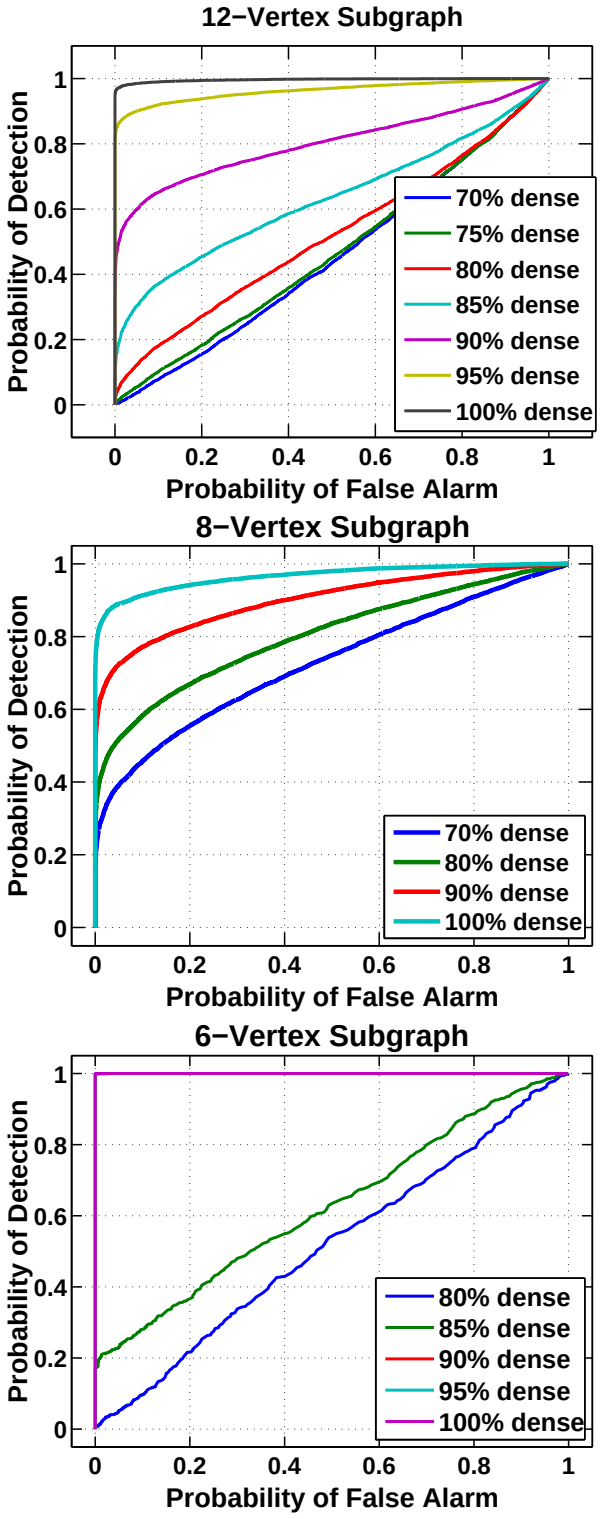


Figure 1. Performance of increasingly complex subgraph detection algorithms. In similar background graphs, a chi-squared statistic computed in the principal 2 components of the residuals can detect a 12-vertex subgraph (top), while the L_1 norms of the top 100 eigenvectors reveal the presence of an 8-vertex subgraph (middle), and sparse PCA enables detection of a 6-vertex subgraph (bottom).

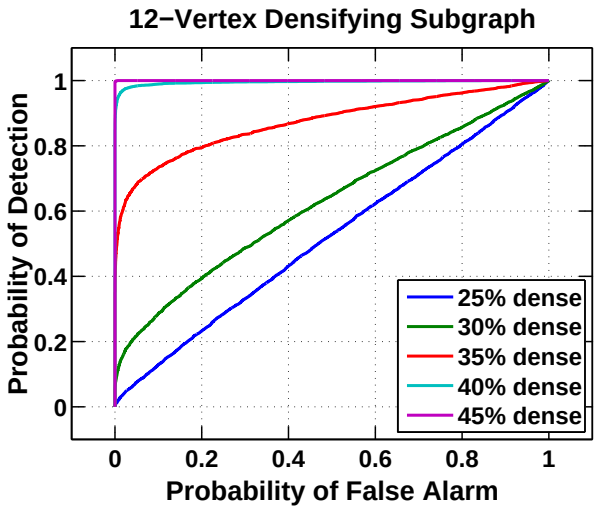


Figure 2. Detection performance in a dynamic graph using a matched filter. Using the chi-squared detection statistic, a densifying subgraph that reaches a density of 40% over 32 time samples is nearly as detectable as a fully connected subgraph in a static background.

the chi-squared detection statistic, a 12-vertex subgraph that is 75% dense is impossible to detect. When knowledge of the dynamics of the subgraph is exploited, however, near-perfect detection is achieved for a subgraph that reaches a maximum density of 40%. Thus, the concept of temporal integration gain is applicable to the subgraph detection problem, just as in other signal processing areas.

This page intentionally left blank.

3. STUDY OVERVIEW

The objective of this one-year study is to build upon the SPG framework, developing new models and algorithms for large, dynamic graphs. Under the internal effort, the techniques were extensively demonstrated in simulation, and on a few relatively small graphs derived from real data. The purpose of the work described in this report is to study the properties of real datasets, preferably with many millions of vertices, and have these properties inform new algorithms that fit into the general SPG setting.

Two datasets were prepared, each of which contain information on entities and their relationships. One dataset is a snapshot of Thompson Reuters' Web of Science, a document database that contains a variety of possible graphs (e.g., citations between documents, coauthorship between people). The other dataset is a web proxy log, which contains the connections made between an organization's internal computers and external web servers. In Section 4, we detail the datasets and the D4M architecture used to store the data.

Based on graphs derived from these datasets, several algorithms were developed to model the data using features other than degree (the feature used for the modularity analysis described in Section 2). Properties of the large graph datasets, such as periodicity in the web proxy logs and time-dependent preferential attachment in the citation graph from the Web of Science, informed new algorithms to better model the data, and, thus, improve residuals analysis. In addition, a generalized linear model was incorporated into the framework, to account for vertex and edge attributes when modeling the probability of edge occurrences. These models are outlined in Section 5.

When applying detection and estimation algorithms to large graphs, being able to evaluate performance objectively, consistently, and efficiently is important. Thus, an additional task under this effort is to document a set of metrics to be used for performance evaluation. These metrics are presented in Section 6. Section 7 demonstrates the performance of the algorithms, on both simulated graphs and graphs derived from the two real datasets, according to the metrics chosen under this effort.

This page intentionally left blank.

4. DATASET PREPARATION

The purpose of this task was to prepare two large relational datasets for processing and analysis. These datasets were to include temporal information, as dynamic data are of specific interest, as well as additional vertex and/or edge metadata. Under this task, raw data is to be preprocessed in a way that allows time-varying graphs to be constructed. New generators are also to be developed for the purpose of Monte Carlo simulations within MIT Lincoln Laboratory's (MIT LL) SPG detection framework.

Under this task, MIT LL received from IARPA one dataset in the form of a large XML document. This document contained data derived from the Thompson Reuters Web of Science database, which is a metadata and citation database for papers in the sciences, social sciences, arts, and humanities with 42 million records from 1900 to present. Records within this database are represented from over 12,000 journals and 148,000 conference proceedings and typically include fields such as author(s), title, publication date, type, document IDs for works cited, and may also include a number of other fields, e.g., subject area, institution, keywords, abstract. This dataset was parsed and ingested into Accumulo triple store database in the D4M (Dynamic, Distributed, Dimensional Data Model) format [16]. D4M is a compact, composable associative array implementation in Matlab (or Octave) developed at MIT LL. This format extends the notion of associative arrays, which are indexed by string keys rather than integers, to large-scale distributed databases, taking the form of sparse matrices, as shown in Figure 3. D4M uses an exploded schema, which pushes the values stored in a database into the rows and columns of a large sparse matrix, and this allows easy computation of graphs using matrix algebra on the extracted arrays, as demonstrated in Figure 4. As shown in the figure, a coauthor graph can be constructed by first extracting an array from the database where the rows are of the form `'docid/<identifier>'` and the columns are of the form `'author/<name>'`, then taking the outer product of this array with itself. Using this database and storage format, MIT LL derived time-varying graphs based on coauthorship (with authors as nodes) and citation (with documents as nodes), see Figure 5. As demonstrated by the statistics of the citation graph in Figure 6, this database contains information to create dynamic graphs with many millions of vertices.

The second dataset was derived from web proxy logs, with the collection setup for an initial ingest of 5 days' worth of data shown in Figure 7. The proxy logs contain 18 fields, including source and destination IP addresses, temporal data, and various additional metadata, as described in Figure 8. The data were parsed and graphs were constructed in a similar fashion to the Web of Science data, using source and destination IP addresses as vertex identifiers. Over the course of one month (September 2011), approximately 650 million records were logged, with 9,753 unique source IPs and 215,271 unique servers, as shown in Figure 9. In the web proxy data, there were also known incidents of malicious activity that can be used as ground truth, which will be discussed further in Section 7. The two datasets combined provide a diverse set of properties, as shown in Table 1, which is desirable since any algorithms developed under this effort should be relevant to graphs from a variety of applications.

Interacting with the database presented significant challenges due to the scale of the problems addressed and the complexity of algorithms applied to the data. The triple store format used by Accumulo, in conjunction with D4M, allows extremely fast data ingest (several

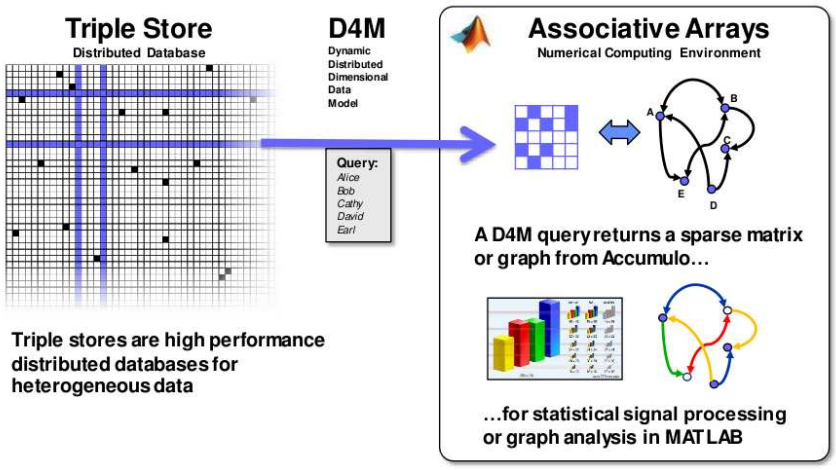


Figure 3. The D4M architecture treats large triple store databases as sparse arrays, allowing easy extraction of graphs from heterogeneous datasets.

thousand inserts per second on each of 20 concurrent processes) and provides an easy interface for users accustomed to working with sparse matrices. However, multiple times, the database crashed and needed to be rebuilt. This appeared to be a resource allocation problem; it seemed to occur mostly when the system memory was being significantly taxed. Memory is also an issue when working with local D4M objects (associative arrays). Adding associative arrays incurs a large memory expense in terms of sorting the row and column keys in Matlab. Working with large graphs is pushing these state-of-the-art tools to their limits, and memory management appears to be the principal bottleneck, both when working with the database and with local objects. Moving forward, awareness of these issues will be important, as will working toward potential solutions. In addition, a development of truly parallel D4M-like language and/or API will be necessary as increasingly large datasets are being processed with increasingly complex algorithmic techniques.

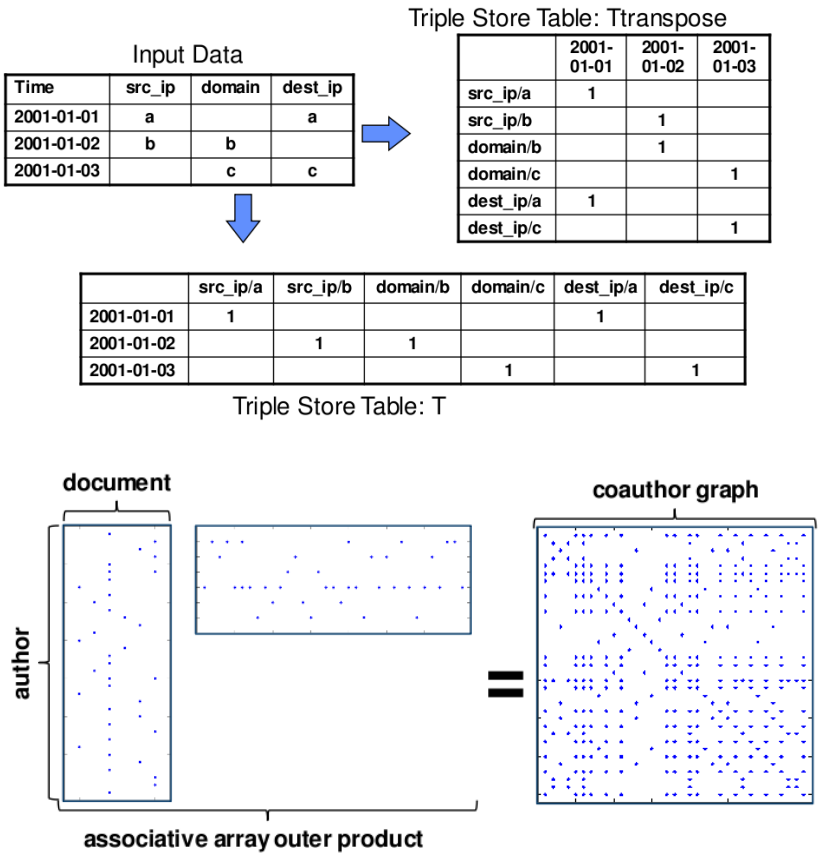


Figure 4. The D4M exploded schema and associative array algebra. In the exploded schema (top), the values in a triple store database are pushed into the row and column labels. To enable fast lookups of both rows and columns, the transpose of the table is also stored. Graphs can be constructed easily from the associative arrays (bottom) using linear algebraic operations on the arrays extracted from the database.

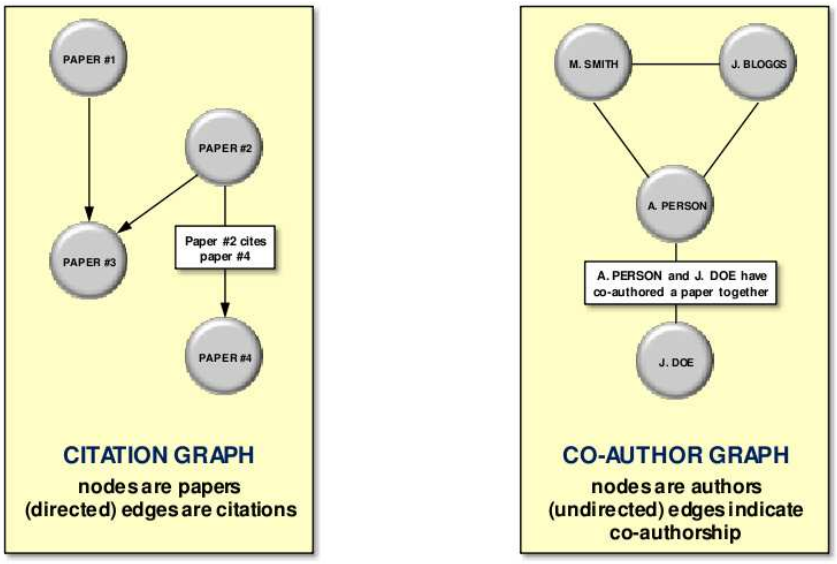


Figure 5. Types of graphs extracted from Thompson Reuters' Web of Science database.

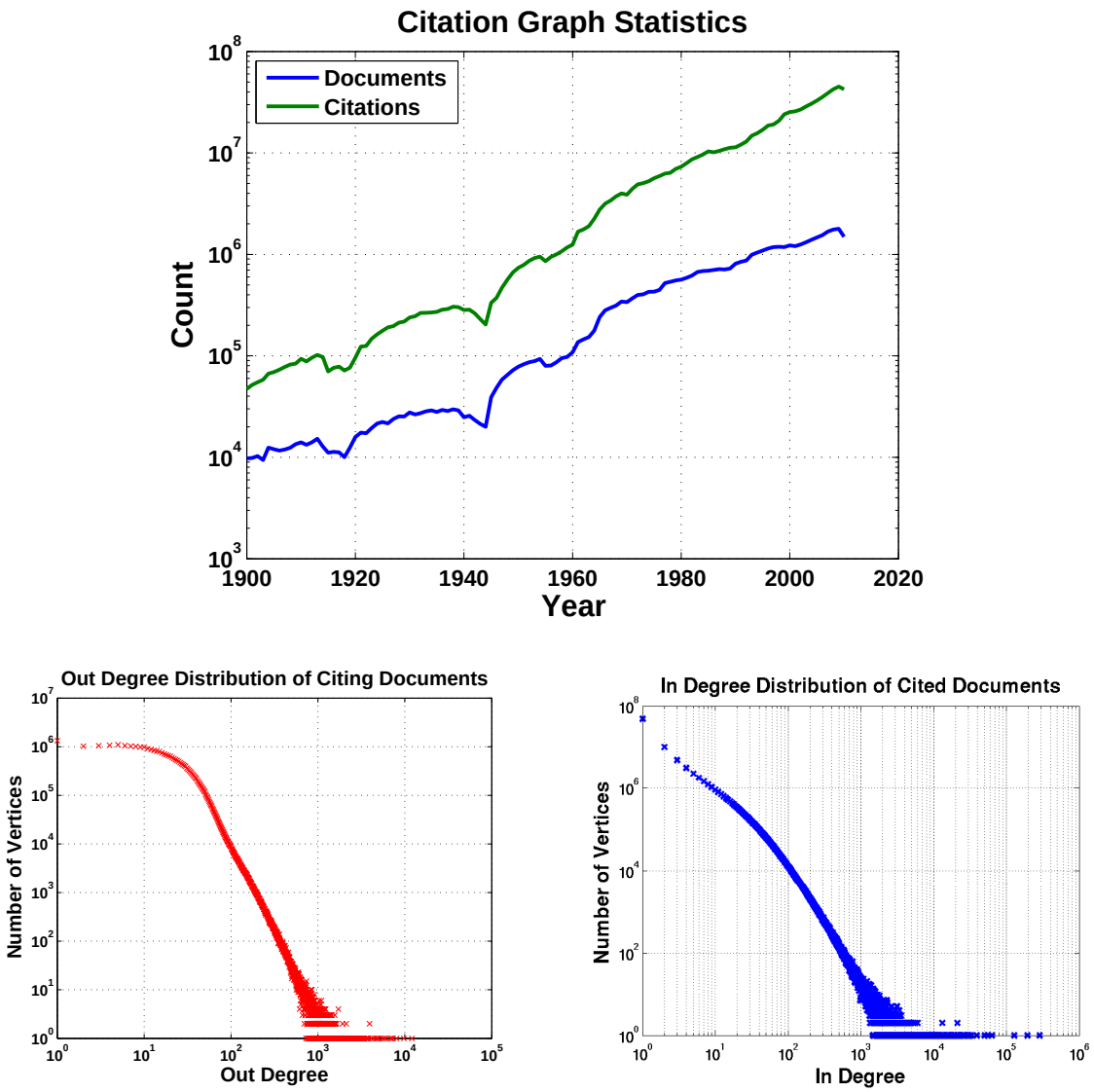


Figure 6. Statistics of citation graphs from the Web of Science database. Between 1900 and 2010, the number of documents per year (top) increases significantly, reaching about 2 million per year in 2009. The number of citations per document also increases over time. Over the entire dataset, the out degree (number of references a document cites) and in degree (number of times a document is cited) both follow power-law-like distributions, as shown in bottom left- and right-hand plots, respectively.

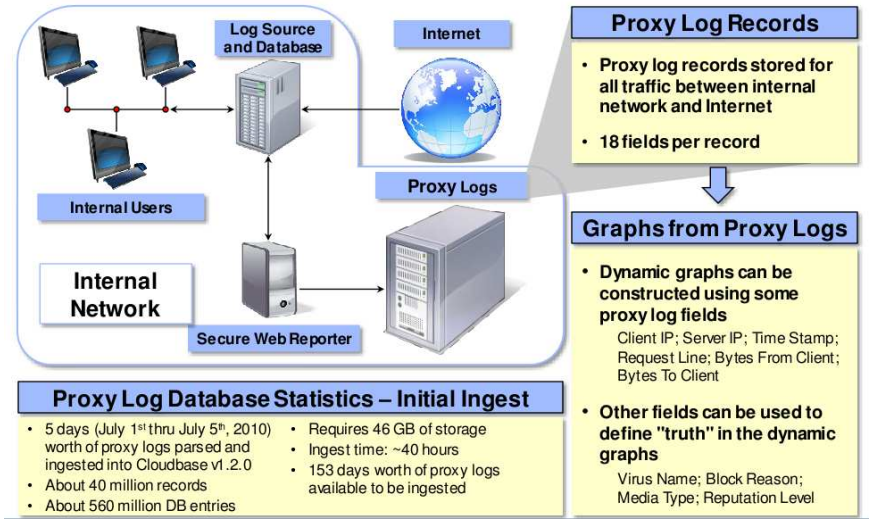


Figure 7. Web proxy data setup and early ingest statistics.

Header	Description	Example
src_ip	Client IP address	128.0.0.1
server_ip	Destination IP address	208.29.69.13
auth_user	Client user name	-
time_stamp	Time of request	[01/Jul/2010:09:00:31 -0400]
req_line	Request line	GET http://google.com HTTP/1.1
status_code	HTTP status code	200
bytes_from_client	Number of bytes sent from the client (in kb)	590
bytes_to_client	Number of bytes written to the client (in kb)	999
referer	URL	http:
user_agent	Client user agent	Mozilla/5.0 (X11; U; Linux x86_64; en-US) AppleWebKit/533.2 (KHTML, like Gecko) Chrome/5.0.342.9 Safari/533.2
attribute	URL Categories	ns, ia
block_res	Filtering action/Reason to block proxy request	-
media_type	Content-type header	application/vnd.google.safebrowsing-chunk
profile	Server profile/policy	-
elapsed_time	Time to process request	0.523
virus_name	Name of virus for HTTP request	-
rep_level	The web reputation of the URL	Neutral
cache_status	Requests on the HTTP port	TCP_MISS

Used to define vertices and edges in the network

Figure 8. Fields from web proxy logs and example entry values.

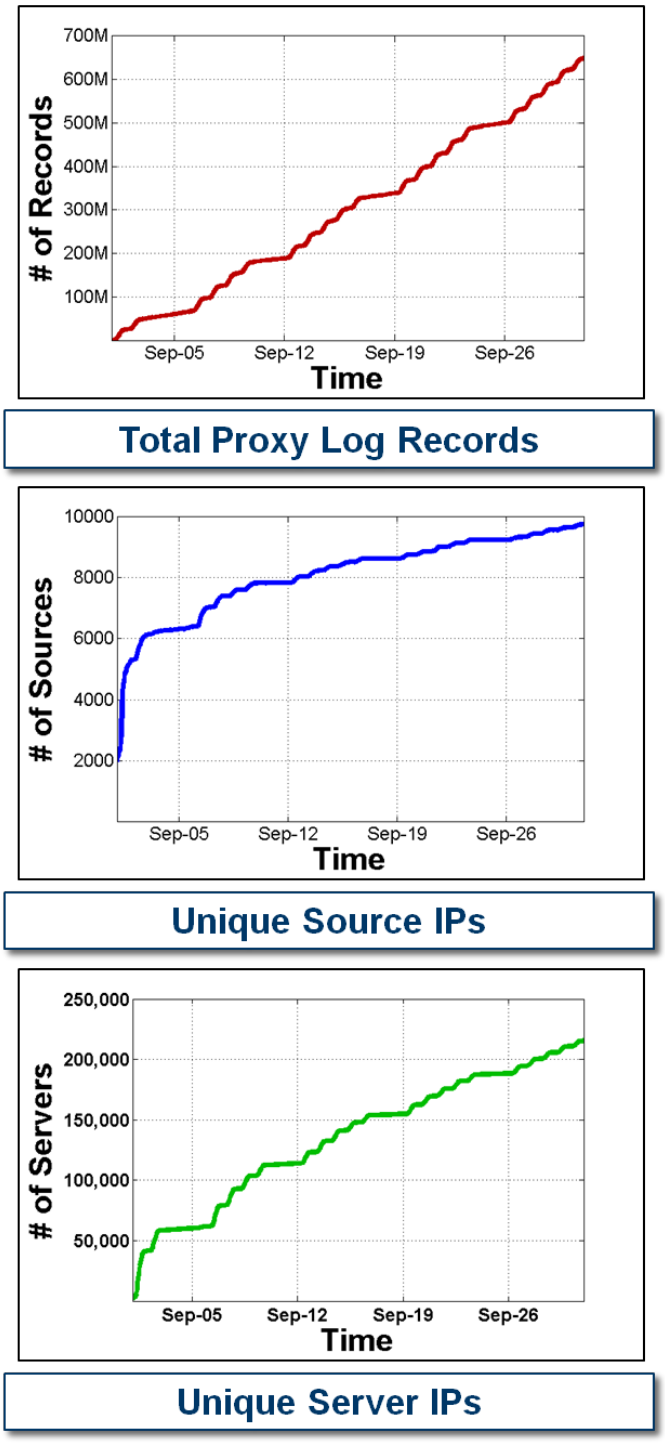


Figure 9. Statistics from web proxy data over the course of one month. The number of records (top) grows steadily over the 30 days, while most of the source IP addresses (middle) are encountered within the first week. The number of server IPs (bottom) also grows more slowly as time progresses.

TABLE 1

Properties of graphs derived from the two datasets.

Property	Web of Science: Citation	Web Proxy Logs
Number of Vertices	42 million	225 thousand
Number of Edges	500 million	650 million
Relationship	"cites"	"connects to"
Time Resolution	1 year	1 second
Bipartite	no	yes
Directed	yes	yes
Attributes	categorical	continuous
Ground Truth	unknown	known events

5. ALGORITHM DEVELOPMENT

The purpose of this task was to expand upon previous algorithmic efforts in the area of uncued anomaly detection in dynamic graphs, and develop new algorithms informed by datasets prepared under this effort.

Under this task, MIT LL developed two new algorithms to expand the previous modularity-based analysis to include modularity of directed and bipartite graphs, as citation graphs are directed and graphs from web proxy data are bipartite and directed. In addition, three new behavioral models were proposed, with corresponding algorithms for anomaly detection. The Web of Science data exhibited a preferential-attachment-like behavior, but one that depended on the recency of the attachments, not simply the accumulated number of connections. The correlation between citation records of different years and a least-squares fit of the current year's citation counts to those of the previous five years, shown in Figure 10, demonstrate this phenomenon. A preferential attachment model with memory was developed that accounts for this behavior, documented in [3].

Also, since the Web of Science is rich with metadata, a generalized linear model (GLM) was developed in which edge probabilities are a function of a linear combination of metadata. This approach leverages the fact that an attributed graph can be created with additional information about nodes and edges, which could potentially enhance our ability to perform inference on graphs. Specifically, for the citation graph, a model where the probability of an edge existing between two vertices (that is, the probability that one document cites another) is determined by the logistic function

$$g(x) = \frac{1}{1 + \exp(-x)}$$

applied to the weighted sum of categorical and real-valued attributes for each pair of vertices: the subjects of the source and target vertex of the pair (categorical) and a real-valued constant associated with each vertex. This technique fuses multiple random graph models and provides a flexible approach to model graphs of various types with additional side information.

In the web proxy data, significant periodic behavior was observed, predominantly due to automated connections. This is exemplified in the connections made by a single server over the course of a day shown in Figure 11. To address this issue, a model based on the weighted average of previous connections was developed. The expected value of the current graph is estimated as the weighted average of previous connections, i.e.,

$$\mathbb{E}[G(t)] \approx \sum_{m=1}^{M} h_m G(t-m).$$

Over all source–server pairs, a histogram of time differences between consecutive connections is plotted in Figure 12, as well as the filter coefficients when fitted over 1 hour by minimizing

$$\|A(t) - \sum_{m=1}^{60} h_m A(t-m)\|_F \tag{1}$$

over all values of h, where $A(t)$ is the adjacency matrix of the web graph at time t with a time resolution of 1 minute. Note that the larger values in the filter occur at intervals that a

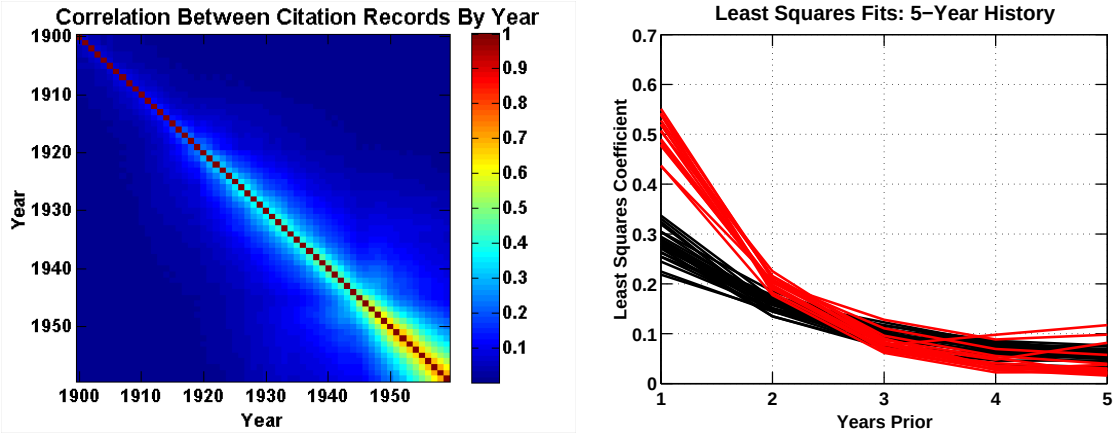


Figure 10. Correlation of year-by-year citation records and a least-squares fit to previous citation counts. The correlation heatmap (left) demonstrates the temporal dependence in citations, and a least-squares fit of current citation counts to those in the last five years (right) confirms this. Similar shapes exist in the year-by-year least-squares fits before 1945 (black) and since 1945 (red), with a phase transition in between.

developer would be likely to hard-code as time to refresh a page (e.g., 10 minutes, 15 minutes, 30 minutes).

Computational complexity was the principal issue dealt with in the algorithm development task. While modularity has a structure that is exploitable for residuals analysis (i.e., the residuals matrix of a sparse graph is a sparse matrix plus a rank-1 matrix), this structure is not always present. Of the three new models, preferential attachment with memory has this form. For the other two models, exploitable structures were found and leveraged. For the moving average filter for adjacency matrices, the expected value term is sparse and can be computed efficiently, and even held in memory in its entirety. Also, while a generalized linear model (GLM) can be arbitrarily complex, if attributes can be decomposed into categories (low rank), attributes of individual vertices (also low rank), or attributes based on connection history (either sparse or low rank), then a useful structure can be exploited to aid in the estimation and analysis process. Exploiting the structure of the models yielded algorithms with complexity linear in the number of edges. When evaluating algorithms in the context of billions—or potentially trillions—of nodes, probabilistic structures that can be exploited for efficient computation will be of very high importance. Analysis of real data highlighted the need for deeper study into effects of uncertainty on algorithmic and modeling performance. While dynamic techniques allow for smoothing some of the noise, considering uncertainty in signal processing context is likely to provide significant insight into this issue.

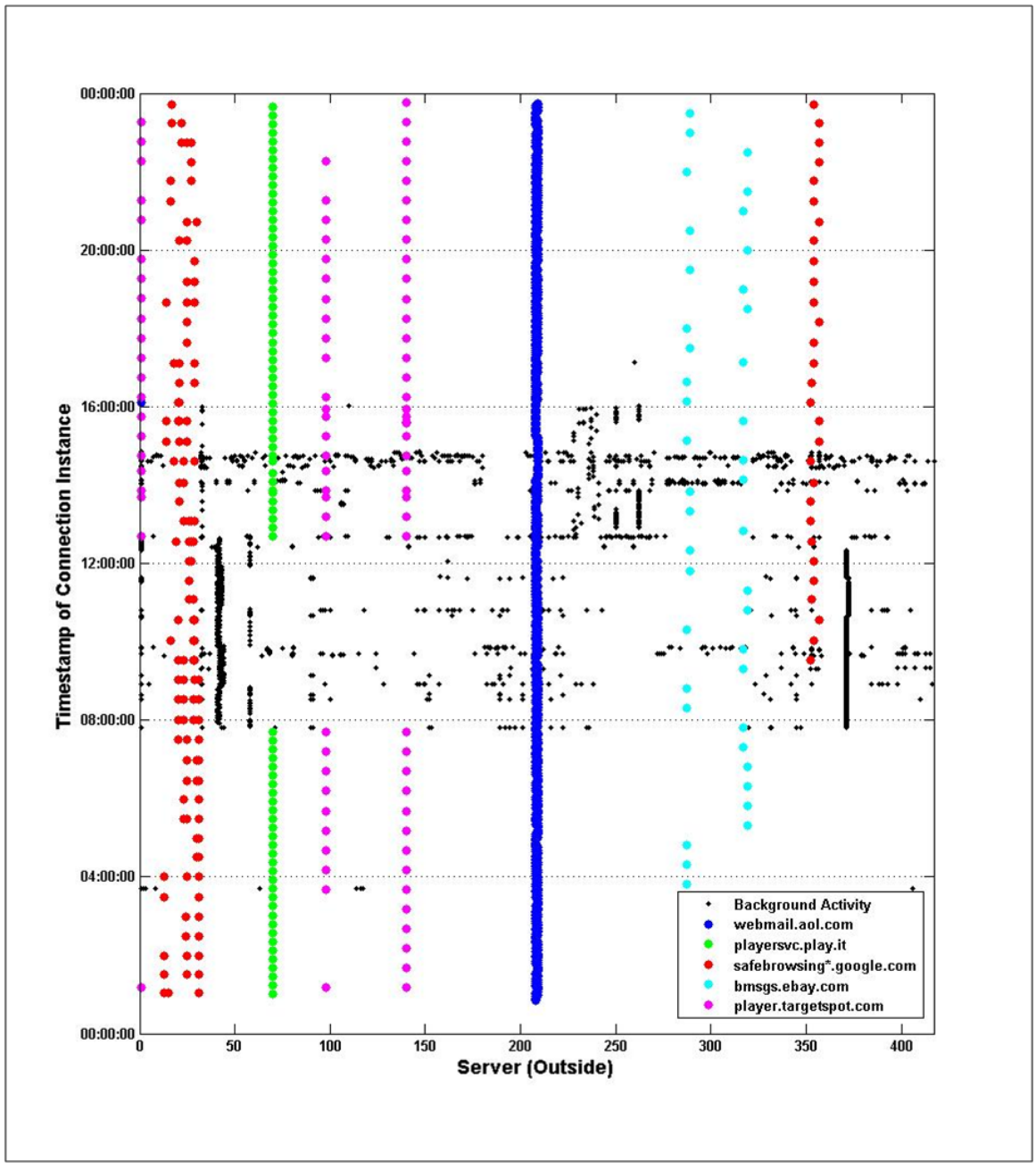


Figure 11. Server connections for a single source. Periodic connections are highlighted in the legend.

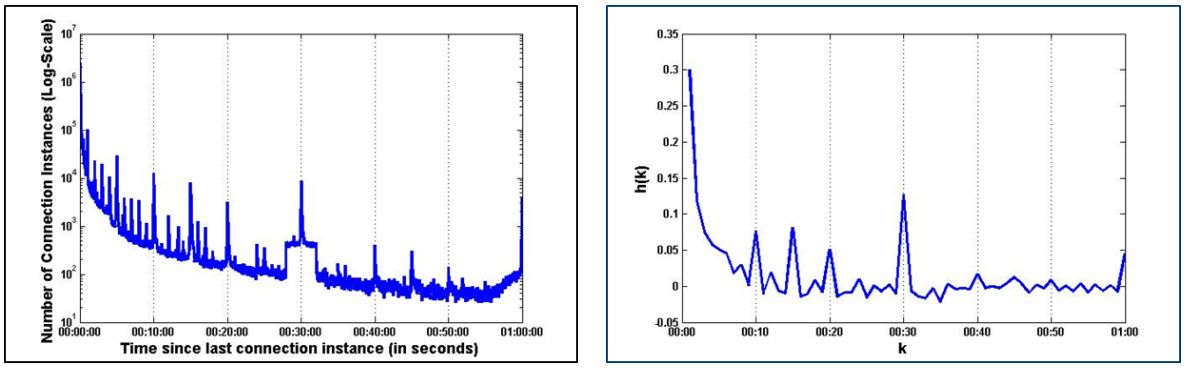


Figure 12. A histogram of times between connections in a web proxy graph (left), and the associated moving average filter to predict new connections (right).

6. METRIC DEVELOPMENT

The purpose of this task was to build a set of metrics used for algorithm evaluation. These metrics should quantify performance both in terms of modeling and detection ability, as well as algorithm scalability to extremely large datasets.

Under this task, metrics in three distinct categories were compiled: detection metrics, which quantify detection performance in the presence of truth (or significance of detected subgraphs in its absence); estimation metrics, which evaluate the quality of a model's fit to the data; and complexity metrics, which quantify the time and memory requirements of an algorithm both empirically and theoretically. These metrics are listed in Table 2.

Issues involved in the development of metrics primarily extended from complexity issues in algorithm development, i.e., the ability to apply metrics efficiently using the new models. While detection metrics are generally easy to apply to a variety of algorithms and datasets, and computational complexity can be computed for any algorithm, estimation metrics are often inherently tied to a particular type of algorithm or model. That is, while it may be feasible to compute certain estimation metrics for some datasets or algorithms, it may be difficult to apply all estimation metrics to all cases. For example, in one model it may be computationally tractable to compute a graph's likelihood, but not the spectral norm of its residuals matrix, and in another model the opposite may be true. Thus, when comparing algorithms, it will be important to avoid making apples-and-oranges comparisons and find a common point of comparison for all algorithms (the ability to efficiently compute estimation metrics will likely be an important algorithm evaluation criterion). This also underscores the need for methods to efficiently characterize datasets in a way that allows performers and evaluators to understand the implications that the data have on the evaluation process.

TABLE 2

Candidate detection, estimation, and complexity metrics.

Metric	Description	Comments
Probability of Detection	Given a threshold, the proportion of detection statistics that fall beyond the threshold when the signal is present	Used when truth is available; Sweep threshold to generate receiver operating characteristic; Aggregate statistics such as equal error rate and area under the curve provide algorithm comparisons
Probability of False Alarm	Given a threshold, the proportion of detection statistics that fall beyond the threshold when the signal is absent	
Statistical Significance	Likelihood of finding a given subgraph (or a subgraph with the same properties) via random sampling of the graph	Used when truth is not available
Spectral Norm of Residuals	Largest singular value of the difference between the adjacency matrix and its expected value	Linear algebraic metrics for closeness of an observation to its expected value; Used for model fitting and as detection statistics
Frobenius Norm of Residuals	Sum of squared residuals across all pairs of vertices	
Graph Likelihood	Likelihood that the observed graph would occur under the assumed model	Used for parameter estimation when fitting data
Reduced Chi-Squared Statistic	Average (normalized) squared distance of observation from the expected value	Used to compare fits of different models to data
Variance of Parameter Estimates from Samples	Measure consistency of parameter estimates when trained on different portions of the data	Used when the model is fit using a subset of the data
Asymptotic Running Time	A tight bound (or upper and lower bounds) on the order of growth of the algorithm as dataset size increases	Big-O (or big-Theta or big-Omega) bound on complexity (see [17])
Empirical Running Time	Wall clock time to completion	
Degree of Concurrency	A measure of dependency between portions of the algorithm	Measures "parallelizability" of the algorithm
Parallel Speedup	Factor that empirical (or asymptotic) running time improves when parallelized	

7. DEMONSTRATION

The purpose of this task was to demonstrate and evaluate the algorithms presented in Section 5, using metrics documented in Section 6, run on datasets outlined in Section 4. This section outlines the experiments run for this demonstration.

7.1 MODULARITY ANALYSIS OF WEB OF SCIENCE GRAPHS

After extracting the first two million records (chronologically ordered), the integrated modularity of the citation and coauthor graphs was analyzed. At this point in time (the years from 1900 to 1959), the citation graph consisted of 4,668,824 nodes, including documents in the database and those cited by the included documents, and 549,726 unique authors [1]. The modularity matrices were integrated as in Section 2 over a five-year window for each of the graphs. The top 30 eigenvalues for each graph are plotted in Figure 13. In each plot, one five-year window has a significant increase in several of the eigenvalues, as indicated in the figure, and these windows were considered further.

The eigenspace of each graph for the indicated years contained some clutter, including high degree vertices in the citation graph and large cliques in the coauthor graph. In addition to these less-interesting phenomena, however, emerging clusters were detected. Sparsity patterns of the adjacency matrices of these subgraphs are shown in Figure 14. In the citation graph, on the top in the figure, two subsets emerge with significant internal connectivity. These documents are primarily in the areas of biochemistry and microbiology, and focus on the metabolic properties of acids and proteins. This subgraph increases its internal connectivity gradually over the window, and the temporal integration applied to the modularity matrices makes the subgraph as strong as a star graph (vertex with extremely high degree) with twice as many vertices. The coauthor graph, on the bottom in the figure, contains two subgraphs whose density also increases over time, in this case pathology case records from Massachusetts General Hospital published in the New England Journal of Medicine, and similar documents in the newly founded American Journal of Medicine. This demonstrates sets of medical researchers taking part in these cases and increasing their connectivity over time. Again, temporal integration strengthens the subgraphs, making them as strong as cliques that are 50%–100% larger.

Since there is no truth available in the Web of Science data, detection performance was evaluated based on statistical significance. In the coauthor graph, one million subgraphs of the same size as the detected cluster were sampled by adding neighbors along a random walk. The samples' integrated densities (weighted sum of volume of the subgraph) are plotted against their normalized integrated densities (where the sequence of the subgraph's volume is made to have unit norm) in Figure 15. Of the one million samples, only 428 fall near the detected cluster (shown by the red dot in the green box), and *all* of the subgraphs within this box stand out in *the same* eigenvector as the detected 32-vertex cluster. Thus, the detected subgraph, coinciding with the inception of a major medical journal, is indeed an outlier among the background of the coauthor graph.

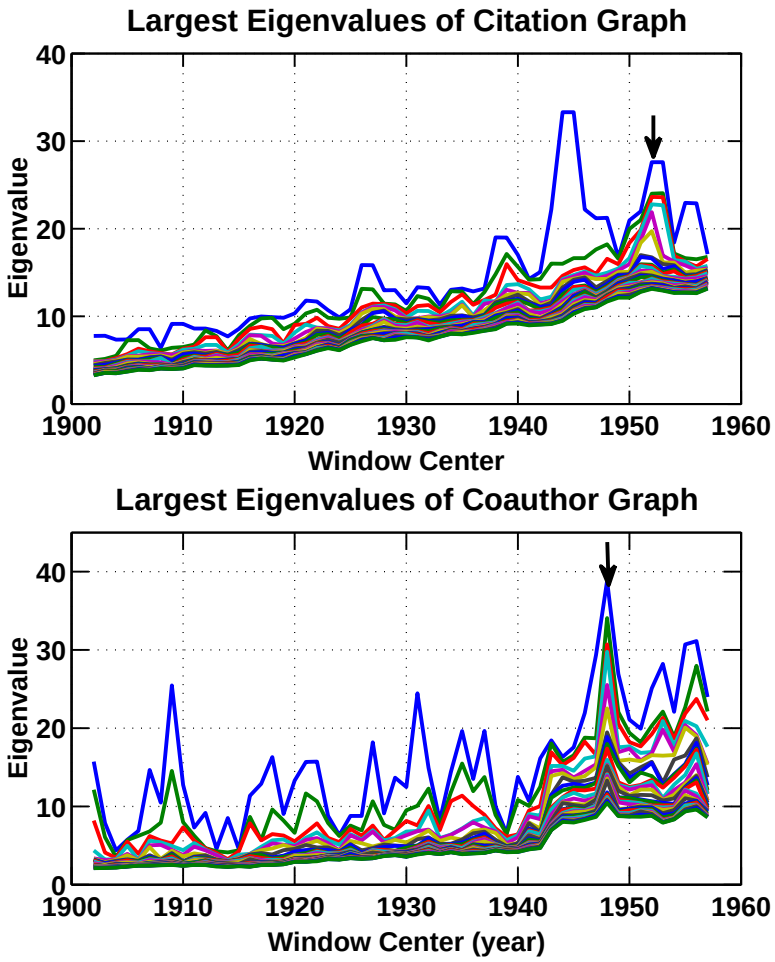


Figure 13. The 30 largest eigenvalues of integrated modularity matrices for the citation (top) and coauthor (bottom) graphs derived from the Web of Science database.

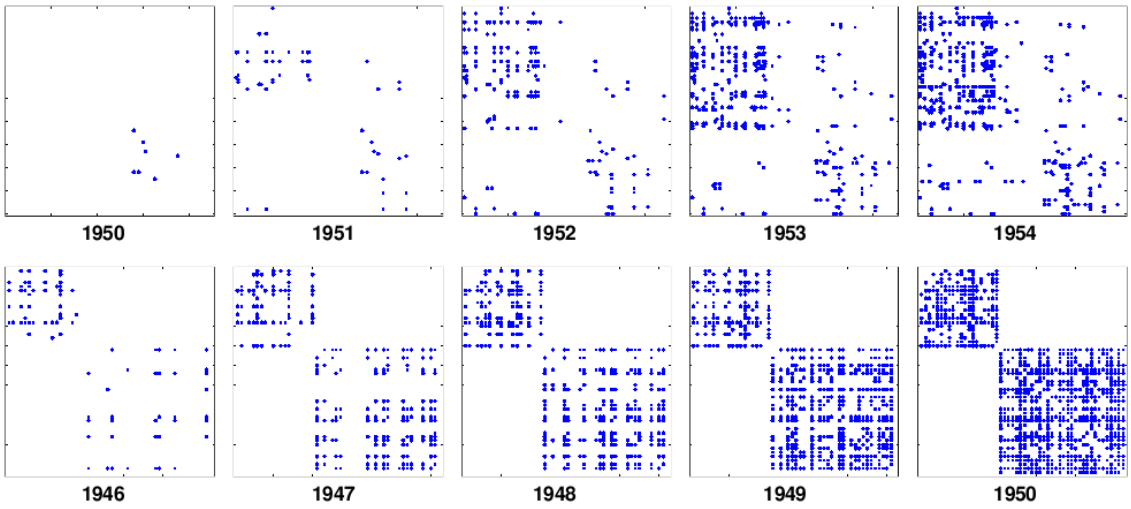


Figure 14. Emerging clusters from Web of Science graphs. The top sequence includes a set of biochemistry documents that gain significant internal connectivity over time in the citation graph, while the bottom sequence shows a subset of medical researchers whose collaboration in pathology cases increases over time.

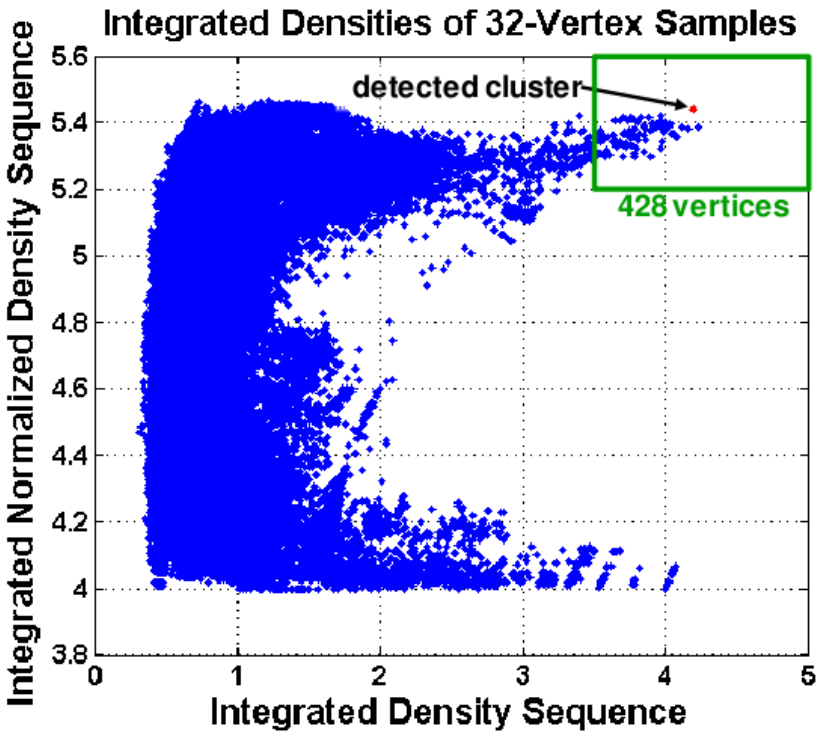


Figure 15. Integrated densities of one million samples from the coauthor graph. A detected cluster is shown by a red dot in the upper right, a clear outlier, and only 428 of the 1 million samples fall within the green rectangle surrounding this point.

7.2 FITTING WEB OF SCIENCE DATA TO PREFERENTIAL ATTACHMENT WITH MEMORY

The Web of Science citation network over the first 80 years was fit to the preferential attachment with memory mechanism referred to in Section 5. This was compared to a standard preferential attachment mechanism, where a document's attachment rate is based on the total number of accumulated citations, and a model where both age and degree are considered to model the citation rate. Results are presented in Figure 16. The values plotted are the normalized variance (reduced chi-squared statistic), given by

$$\frac{1}{|V|}\sum_{i=1}^{|V|}\frac{(k_i-\lambda_i)^2}{\lambda_i},$$

where k_i is the in degree of vertex i and λ_i is its attachment rate. The values for preferential attachment with memory are closer to 1, indicating that it is a better fit to the observed data than the other models.

To evaluate detection performance using a graph with a preferential attachment with memory (PAM) graph, a Monte Carlo simulation was run in which the background is a 2048-vertex graph, generated by preferential attachment with memory over 16 samples. At the last of the 16 samples, a signal may be embedded in which high-rate vertices swap their attachment rates with high-degree vertices. As a detection statistic, the spectral norm of the residuals matrix is used. Results are shown in Figure 17, with detection performance compared to cases using residuals based on modularity, for either the current sample or the accumulated graph. Using the true model improves residuals analysis, and the improvement in detection performance is seen in the receiver operating characteristics. Performing spectral analysis on the Web of Science citation network, fit to preferential attachment with memory with a five-year history, yields singular values as shown in Figure 18. Times to completion for the singular value decomposition are also shown in the figure.

7.3 FITTING WEB OF SCIENCE DATA TO GENERALIZED LINEAR MODEL

The Web of Science data was also fit to the GLM, to demonstrate the impact of using vertex metadata in modeling. Six covariates were used: five indicating the age of the document being cited (0 years, 1 year, 2 years, 3–5 years, and 6–10 years), and one indicating whether or not the two documents share the same subject. The model was trained on a random sample of 5,000 papers published in each year from 1900 to 1959, and 20,000 papers published within the past 10 years (as candidates to be cited). The maximum likelihood estimate for the weights for each covariate were estimated via an iterative procedure.

Estimates of the parameters over the course of the years is shown in Figure 19. The red band in the top heatmap indicates that publishing in the same subject has a drastically greater impact on citation probability than does a document's age. In the center and bottom plots, the co-subject covariate is separated, showing a slight increase in the importance of the recency of the cited document in 1945 (the year where the phase transition occurred in the preferential attachment coefficients), as well as a decrease in the importance of the document

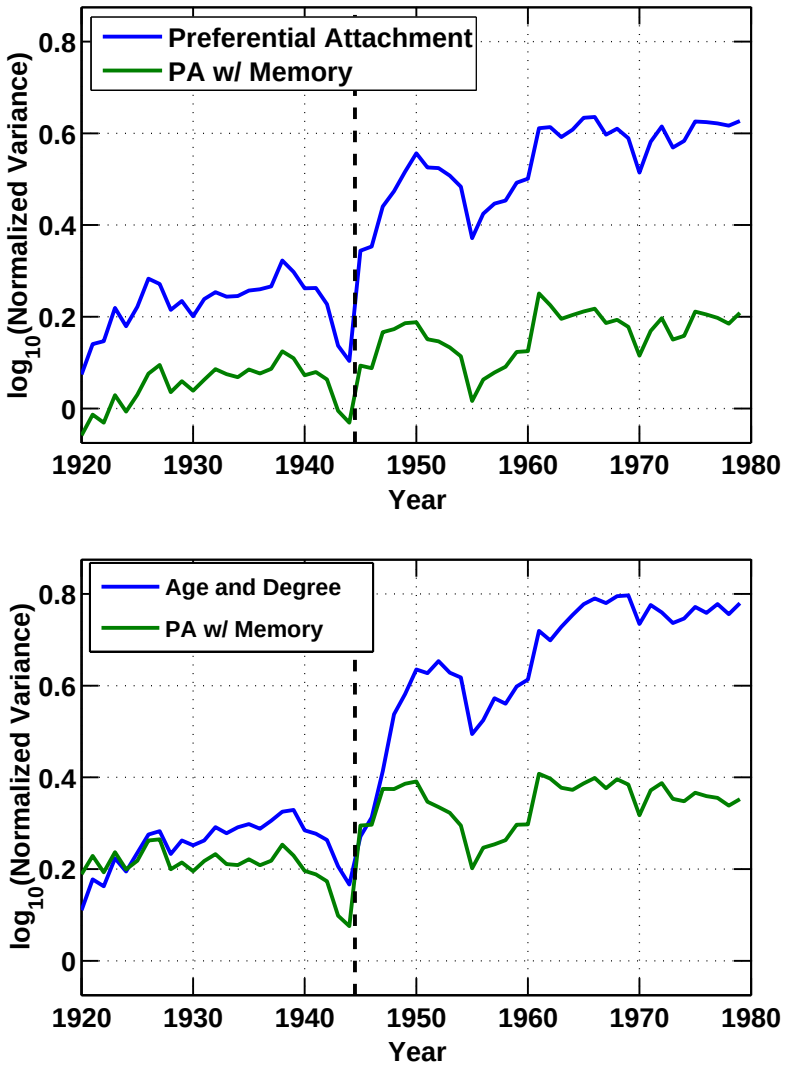


Figure 16. Comparison of model fits to Web of Science citation network. The reduced chi-squared statistic is plotted. In the top plot, the preferential attachment model with memory is compared to a standard preferential attachment mechanism, using all previous attachments to generate new attachment rates. In the bottom plot, preferential attachment with memory is compared to a model in which age and degree are both considered, using the attachments that occur within the documents in the database.

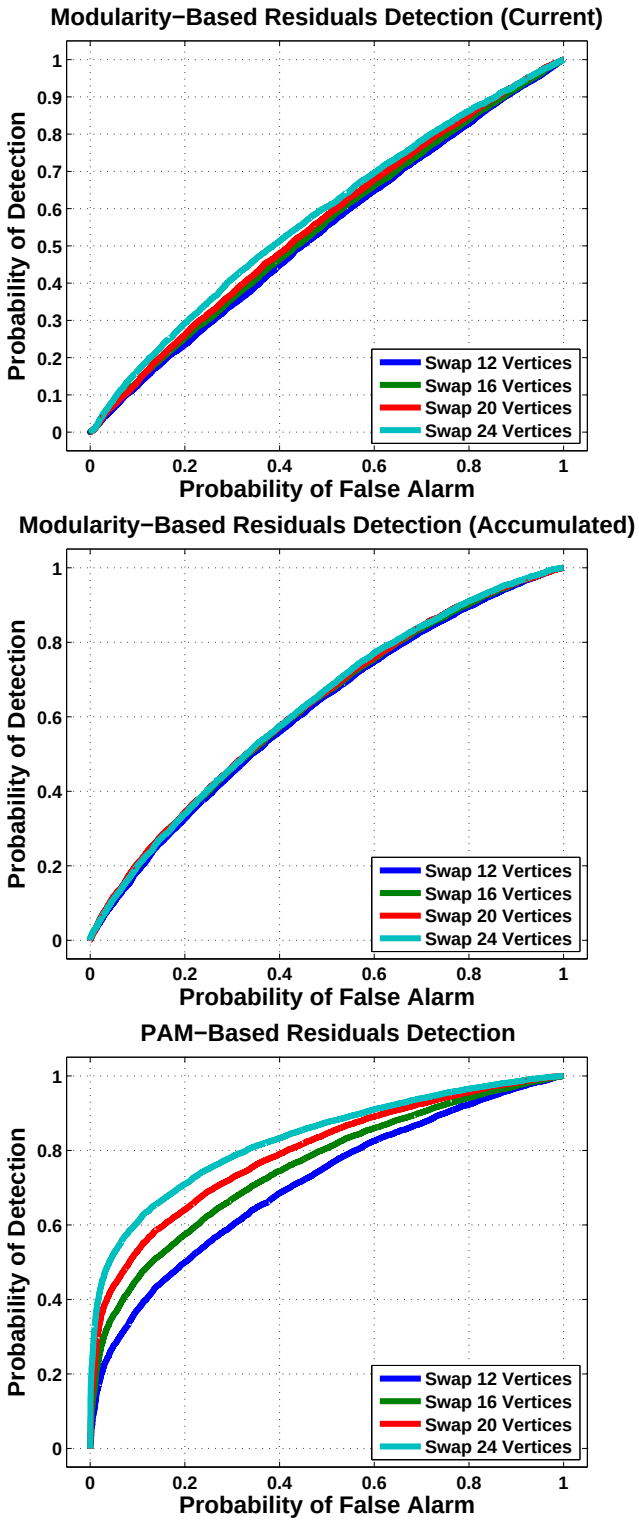


Figure 17. Detection performance in a background that grows by preferential attachment with memory. Performance is significantly improved when the observed graph is fit to a preferential attachment with memory model (bottom), rather than using modularity (top and middle).

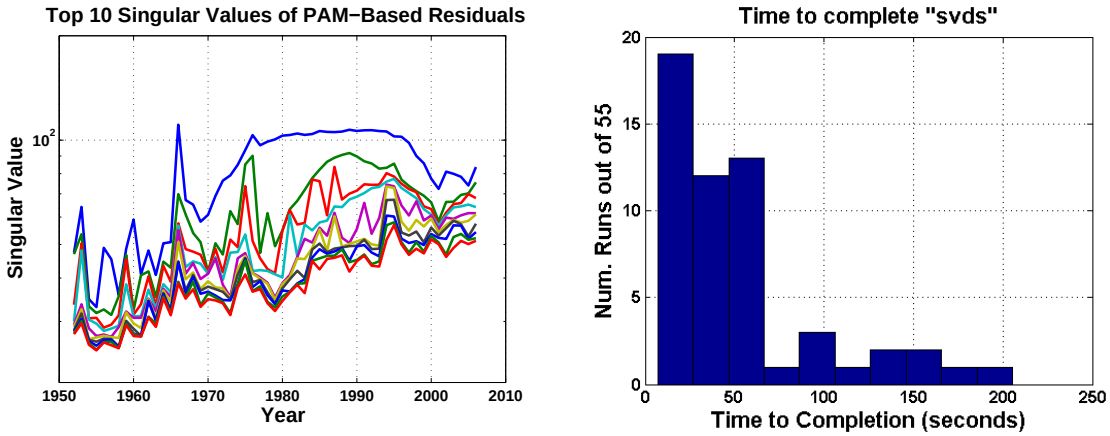


Figure 18. Spectral analysis of Web of Science residuals after fitting to preferential attachment with memory. The top 10 singular values for each year are presented (left), along with the time required to compute the singular values on a commodity machine (right).

being in the same subject. This demonstrates that metadata in addition to time and degree are useful for probabilistic modeling of large graphs.

Detection performance using backgrounds derived from the GLM was also evaluated in a Monte Carlo simulation. Residuals were analyzed using modularity, the GLM with estimated parameters, and the GLM with true parameters. Results are shown for all cases in Figure 20. In the top row, a 4-vertex clique is embedded into a 5-class random graph with 200 vertices in each class. The probability of a connection between vertices in the same class is 0.0075, and between vertices in different classes it is 0.0025. From left to right, the probability of an intra-subject connection is reduced to 0.005, and then to 0.0025, making the graph an Erdős–Rényi random graph. As the community structure decreases, the cluster becomes easier to detect using the spectral norm of the residuals. Note that performance is the same using the GLM whether parameters are estimated or given. In the bottom row, rather than embedding a clique, 20 of the vertices are given the wrong class label. As the probability of inter-class connection increases from 0.0025 (left) to 0.005 (center) to 0.0075 (right), the mislabeling become more detectable using the same statistic. In all cases, fitting the observation to the GLM yields performance at least as good as, and usually much better than, using modularity. Note here that there is a performance difference between using true and estimated parameters, likely due to additional errors in the estimation process due to the mislabeling.

Finally, the Web of Science citation graph was fit to an approximation to the GLM in which the probability of citation is the product of (1) the citing document's rate of citing others, (2) the cited document's rate of being cited, and (3) the rate at which documents of the same subject as the citing paper cite documents of the same subject of the cited paper (i.e., a unique constant for each ordered pair of subjects). This allows an expected value matrix whose rank is commensurate with the total number of subjects. A residuals analysis similar to the previous modularity-based technique was performed, and five analytical chemistry papers stood out significantly in the residuals space, as illustrated in Figure 21. These documents all have high degree, and to this data have accumulated thousands of citations, but they are

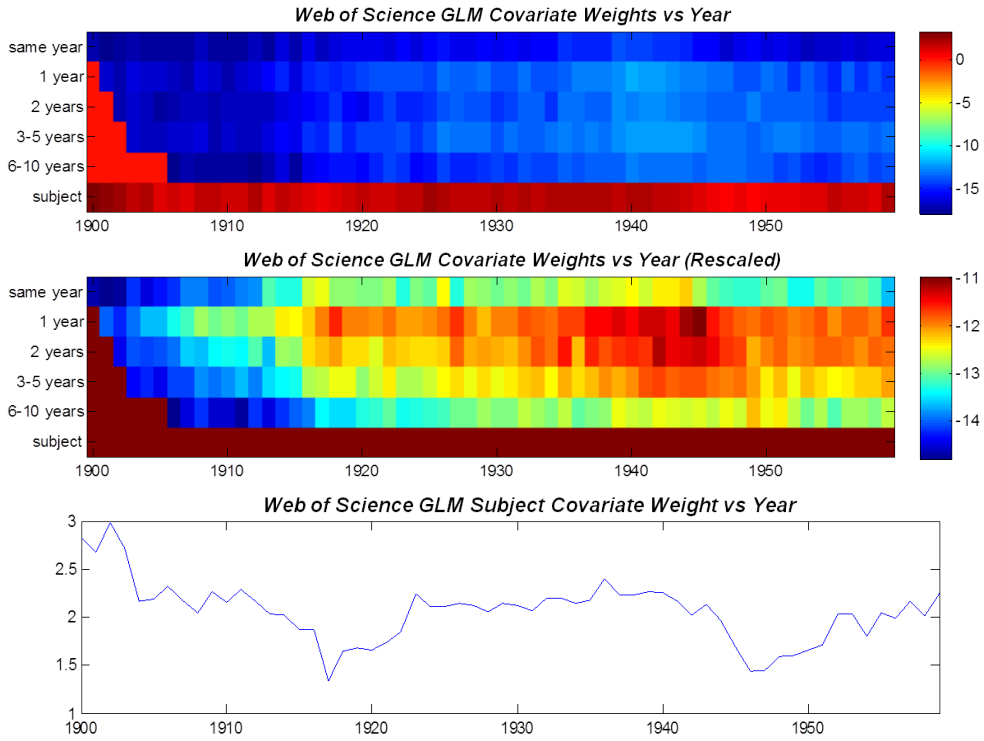


Figure 19. Covariate weights when fitting Web of Science to a generalized linear model (log base 10 scaling). Publishing in the same subject has a much greater impact that publishing recently, as shown in the top plot. The rescaled heatmap in the center, with subjects on the bottom plot, shows a peak in the importance of recency around 1945.

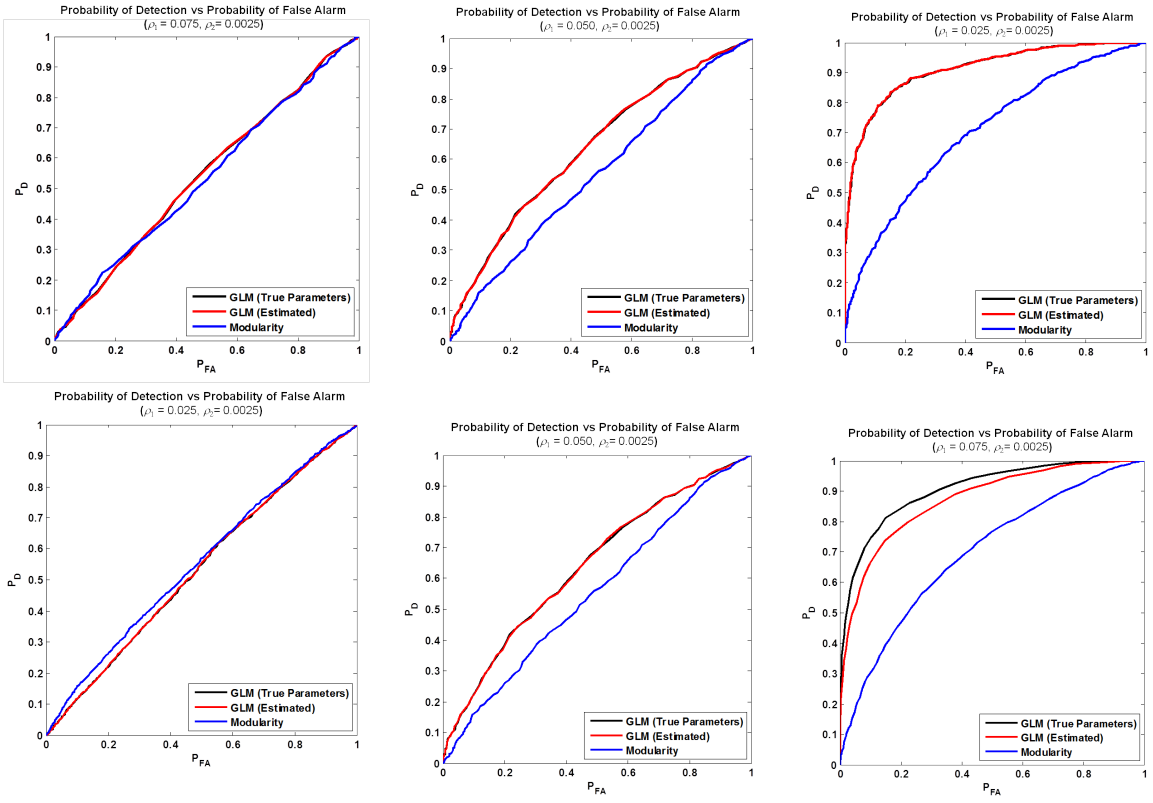


Figure 20. Detection performance in a background generated by a generalized linear model. In a 5-class simulation, the embedding of a 4-vertex clique (top row) is more detectable as the intra-class connection probability is decreased (left to right), while the detection of mislabeled vertices becomes easier as this probability is increased (bottom row, left to right).

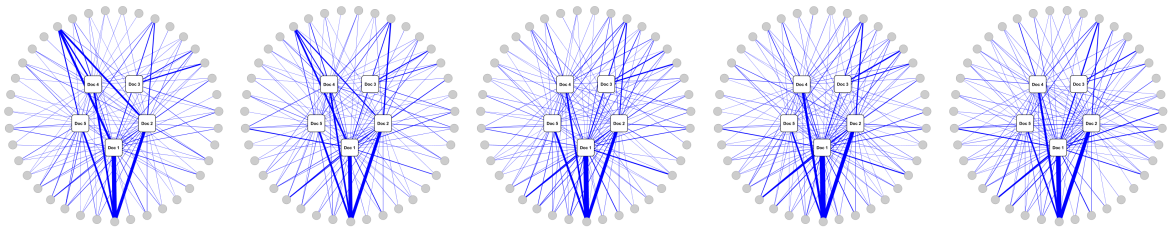


Figure 21. Documents detected through GLM-based residuals analysis. In the years from 1972 to 1976 (left to right), five analytical chemistry papers (squares in the center) stand out in the residuals space due to being cited by many different subjects (circles around the perimeter).

greater outliers than several papers with even greater degree. As shown in the figure, these documents are cited by papers in a wide variety of subject areas, and this significant amount of cross-subject citation boosts their strength. This demonstrates the impact of vertex and edge metadata on detection results in the analysis of graph residuals.

7.4 EVENT DETECTION IN WEB PROXY DATA

After being parsed and inserted into Accumulo in the D4M format, the web proxy logs were used to construct a time-varying graph with a resolution of one minute. A filter was constructed to fit the current connections to the previous one hour of activity according to (1). The ratio of the Frobenius norm of the residuals matrix to the Frobenius norm of the adjacency matrix over the course of one day is plotted in Figure 22. The spike in activity at approximately 0400 is a strong outlier, making the residuals much stronger. Upon further inspection, this point in time coincides with the connection of a computer to a server for operating system updates, which concentrates residuals on a small subset of vertices. The subtraction of periodic behavior via the moving average filter significantly lowers the background activity, thus making this activity much more prominent. As in the citation graphs, taking into account past connections improves the ability to predict new ones, and, thus, enables detection of other behaviors via residuals analysis.

In addition to the aforementioned coordinated behavior, which turned out to be innocuous, there was a known instance of a botnet on one of the internal machines. The modularity of the web proxy graph around the moment the infected computer was connected to the network was analyzed, with the principal two-dimensional subspace defined by the left singular vectors (sources) and right singular vectors (servers) are plotted in Figure 23. Before the infected source is placed on the network, the residuals are presented on the top row. Specifically, in the space of the sources (left), no vertex stands out exceptionally. Fifteen minutes after the computer is connected to the network (bottom row), the infected node is significantly separated from the rest of the sources, and most of the outliers among the servers connect to this vertex. The botnet caused the infected computer to connect to many servers with a small number of incoming connections, and when the residuals space is integrated over 30 minutes, as shown in the figure, this behavior causes the infected source to stand out in the residuals space. This separation occurred after 15 minutes of activity, while the botnet went undetected for 10 days until it was found by manual inspection of logs. This demonstrates the potentially vast

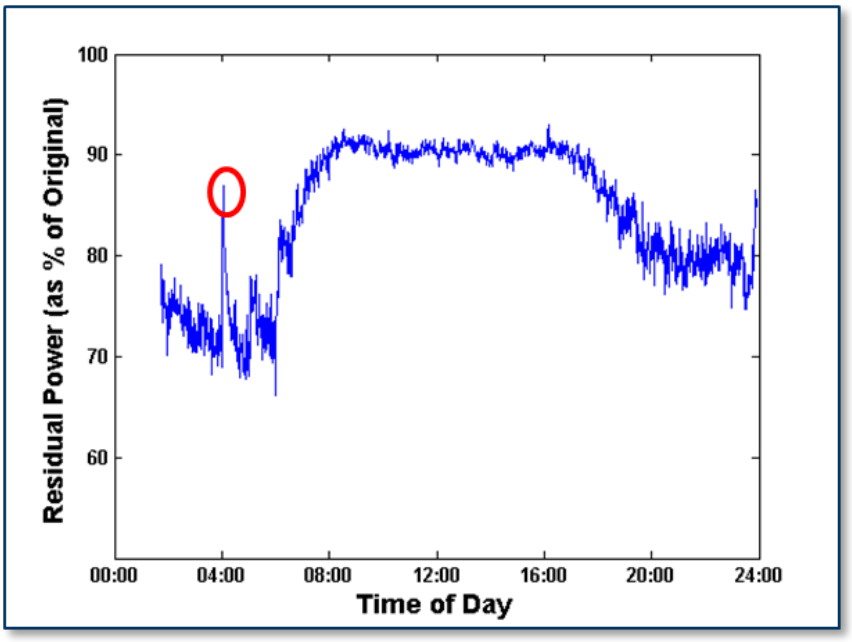


Figure 22. Detection of coordinated behavior in web proxy data using a weighted average of past connections. The highlighted spike in activity corresponds to coordinated activity between a few servers and a single source.

improvement in the detection of coordinated behavior enabled by graph analytics. Overlayed scatterplots of the 60 minutes preceding the moment plotted in Figure 23 are shown in Figure 24, indicating that the this botnet could be detected via analysis of residuals with very few false alarms.

A log from a seized botnet was also used to generate foregrounds which, when embedded into the web proxy graph, allowed Monte Carlo simulations to quantify detection performance in the background. The organization of the botnet is presented in Figure 25. The log contains 44,448,856 records covering 19 days, and includes 667,029 unique source IP addresses (bottom layer) and 10,207 unique repeater addresses. Fifty of the more active source IP addresses were embedded into the web proxy log graph, and their residuals were integrated over a 24-hour window. The embedding procedure caused a substantial difference in the L_1 norm of one of the singular vectors of the residuals matrix, as shown in Figure 26. This analysis is similar to that performed in [11]. In the space of the singular vector with the anomalously small L_1 norm, the sources with the embedded botnet traffic are separated from the other nodes, as also shown in the figure. Running a simulation with 1,315 random embeddings into randomly selected 24-hour windows, and detecting based on the largest deviation in singular vector L_1 norms, detection performance is shown in Figure 27. The equal error rate is just under 20%, which may be reduced if more sophisticated temporal integration is used (the present results use a simple averaging filter). This analysis allows detection performance to be quantified in a rigorous way, with the infected nodes being randomly distributed across the graph.

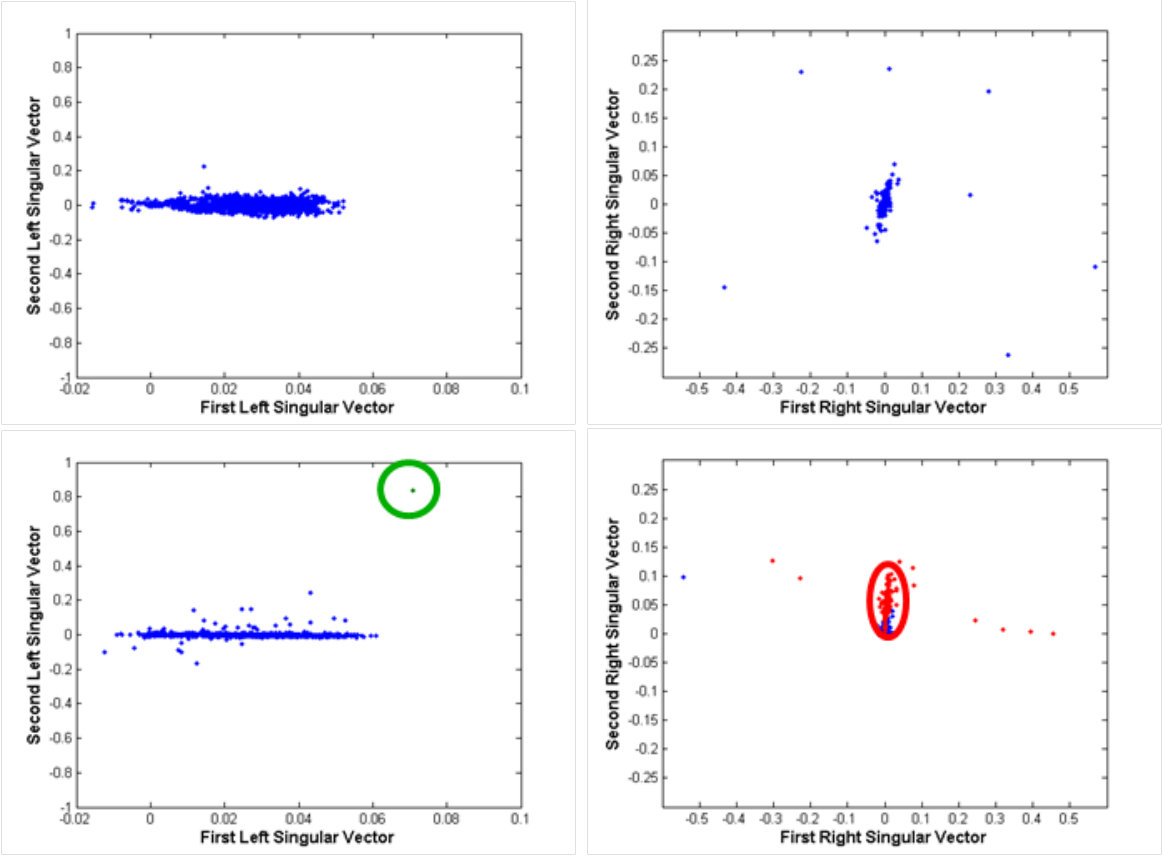


Figure 23. Detection of botnet activity in web proxy logs. After the infected source is connected to the network (bottom row), the infected vertex (highlighted in green) stands out in the residuals among the sources (left column) much more than any vertex does before it is connected (top row). The space of servers (right column) is also primarily dominated by servers to which the infected node connects (highlighted in red).

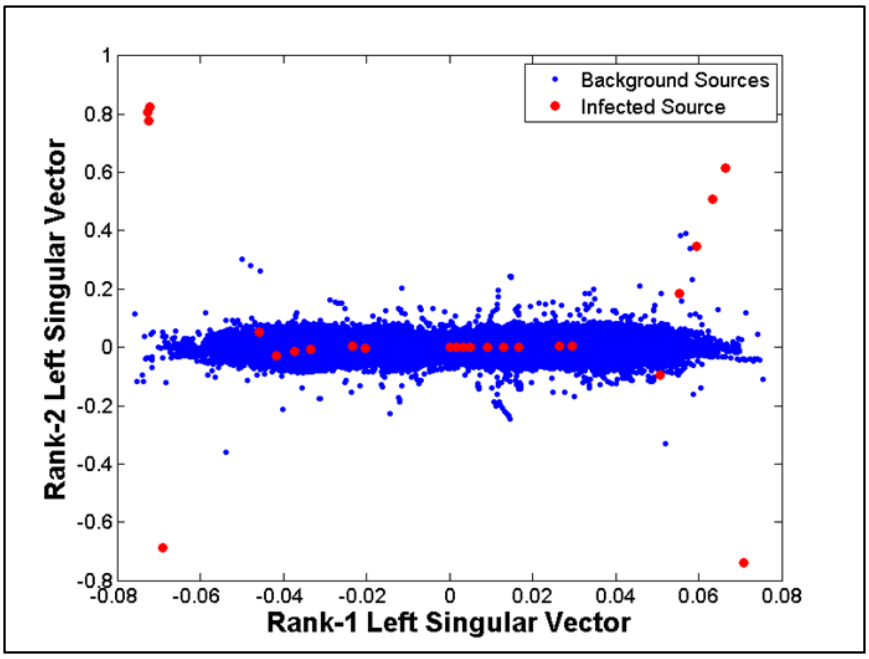


Figure 24. Residuals among sources for 60 minutes before the infected computer is connected. The infected node, highlighted in red, stands out much more than any other node does over this time, suggesting that it can be detected with few false alarms.

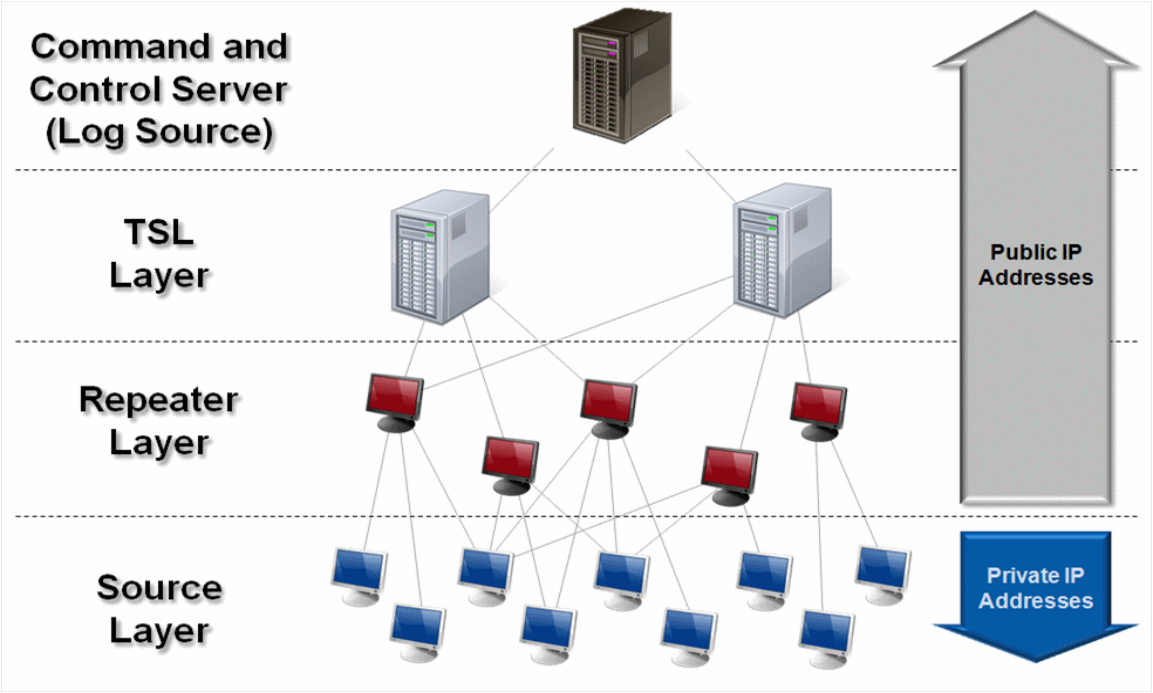


Figure 25. Organizational structure of a seized botnet.

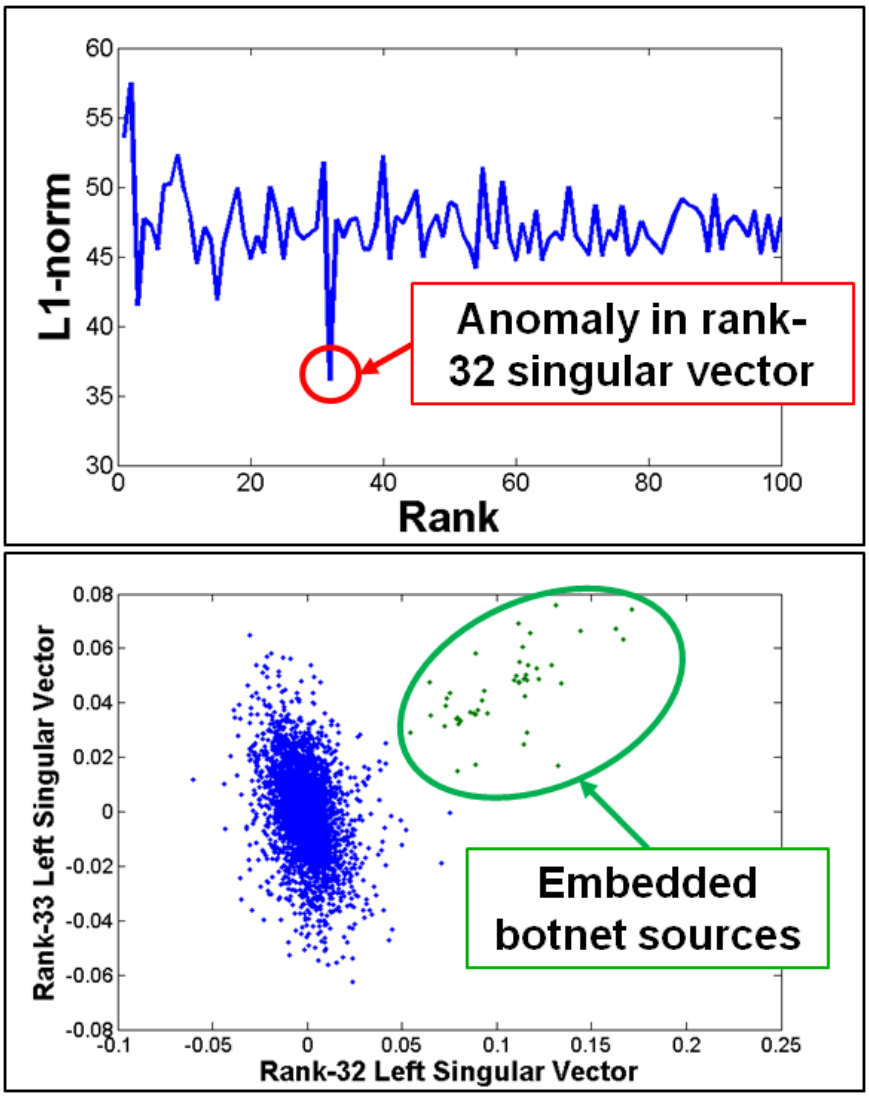


Figure 26. Detection of embedded sample from the seized botnet in a web proxy graph. The embedding of a sample from the seized botnet log into a background graph created from the web proxy logs causes one of the left singular vectors in the residuals matrix to have a very small L_1 norm (top), which indicates that the embedded subgraph is separable in the space of that vector (bottom).

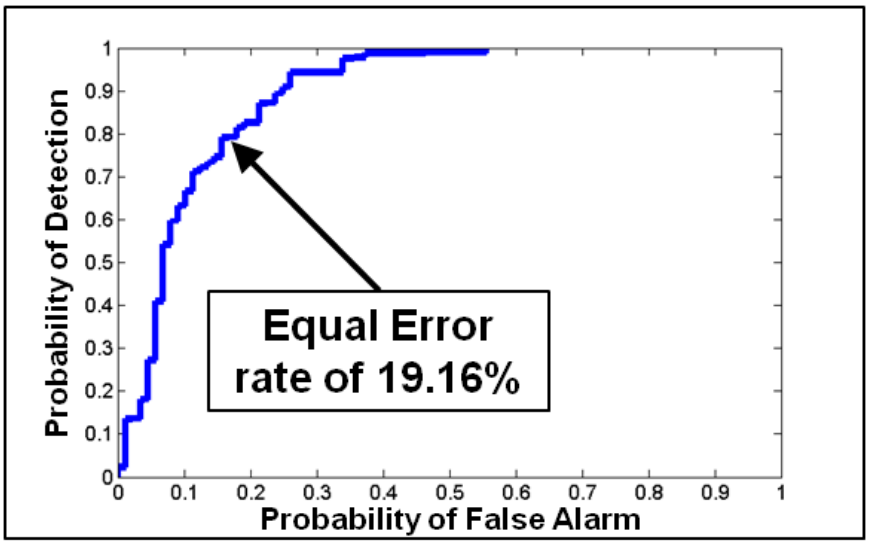


Figure 27. Receiver operating characteristic for the detection of a sample from the seized botnet embedded into a web proxy graph.

7.5 COMPLEXITY ANALYSIS AND DEMONSTRATION AT SCALE

The detection algorithms presented in this report all rely on performing an eigendecomposition on the graph's residuals matrix. Since, in the applications of interest, graphs are typically sparse, k eigenvectors of a graph's adjacency matrix can be computed by an iterative procedure (the Lanczos method) in $O((|E|k+|V|k^2+k^3)h)$ time, where h is the number of iterations, which depends on the smallest gap between consecutive eigenvalues. For most random graph models, however, the residuals matrix is dense. Thus, for computational tractability, it is important that these models have an exploitable structure that allows easy matrix-vector multiplication (the primary driver of the Lanczos method's complexity for small k).

Classical modularity analysis benefits from the fact that the expected value matrix has rank 1, and thus matrix-vector multiplication can be implemented as (1) multiplication by a sparse matrix, (2) a vector inner product, (3) a scalar-vector product, and (4) a vector addition [1]. Preferential attachment with memory has a similar structure, and thus computation benefits from the same technique, with the asymptotic running time not changing from that for the adjacency matrix. The moving average adjacency filter, on the other hand, uses the weighted sum of previous connections to define the expected value. If most connections are not seen over the course of the time window, the expected value matrix will also be sparse, and the complexity will scale with the total number of connections seen over the time period considered.

The generalized linear model does not, in general, have such an exploitable structure. Maximum-likelihood training requires an iterative numerical approach. The cost at each iteration is dominated by p^2 vector-matrix multiplications by the $n \times n$ covariate matrices, for a total cost of $O(n^2p^2)$ per iteration. The complexity of residuals analysis is dominated by the Lanczos-type spectral analysis, which requires p matrix-vector multiplications at each step,

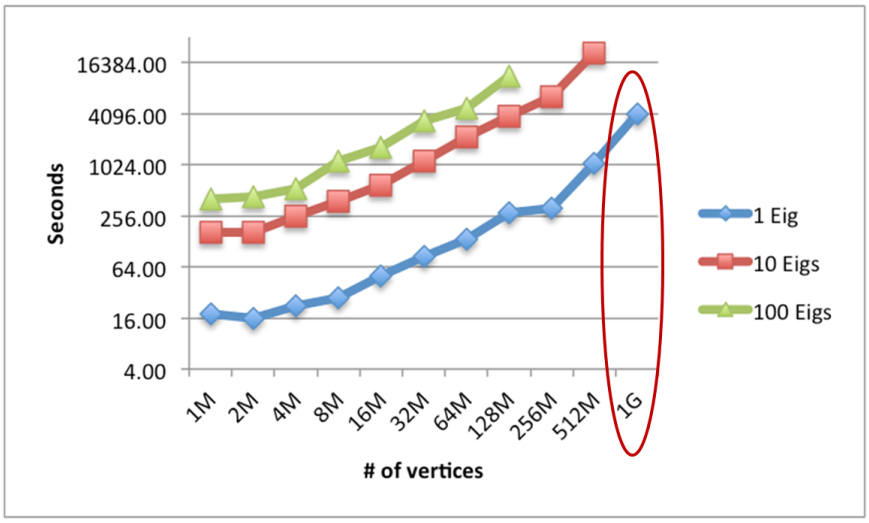


Figure 28. Running times of parallel computation of graph residuals.

at a cost of $O(n^2p)$ per iteration. As with modularity approaches, if the matrix of model parameters has a low-rank structure, we can exploit it to improve computation. For example, assume the p covariates for each edge are "categorical" covariates, i.e., binary indicators of membership in one of p classes. Under this assumption, the sum of all covariate matrices can be represented as a rank-p matrix, reducing the respective costs of training and analysis to $O(np^3)$ and $O(np^2)$ per iteration. Also, exploitable approximations, such as the one used at the end of Section 7.3, and simplified approximations for parameter estimates, will be extremely important in analysis of large networks using metadata for modeling.

To demonstrate that residuals analysis can be performed on graphs even larger than those considered in this study, a parallel eigensolver for modularity-based residuals analysis was implemented in SLEPc [6]. Eigenvectors were computed on simulated graphs, with sizes from 2^20 to 2^30 vertices and an average degree of 8, distributed across 64 commodity machines. Running times to compute 1, 10, and 100 eigenvectors are presented in Figure 28. Computing 1 eigenvector requires far less time than computing 10, while computing 100 requires only slightly more time, due to the relatively large gap between the first and second eigenvalues. On a 1-billion vertex graph, 1 eigenvector was computed in about 70 minutes, and 2 eigenvectors required approximately 9 hours. This result, however, demonstrates that such residuals analysis on gigascale graphs is possible, and can likely be done subject to application-relevant latencies given more resources or possibly custom hardware.

7.6 DEMONSTRATION CHALLENGES

In the process of demonstrating the algorithms on the datasets, the most significant issues were computational complexity and the evaluation of detection performance in the absence of ground truth. Algorithm complexity limited the capability to train the generalized linear model (GLM), although more recent developments have significantly reduced the time required for training (as discussed in the previous subsection). The Web of Science citation

graph was sampled for the purpose of training, and the parameters were cross-validated across several samples. These samples were taken uniformly at random across source and destination vertices for each year. While a more efficient method of training was determined later, in the course of future algorithm research, it is likely that some approaches will have complex training procedures that will require sampling the network. Graph sampling is still an active area of research, and additional work in which extremely large graphs are the data of interest will need to consider sampling procedures developed/applied under the study and generate new techniques.

When evaluating the significance of detection results when truth is not available, sampling is also used. Properties of the detected subgraph (such as the rate of densification, as shown in Figure 15) are compared to those of many randomly sampled subgraphs. Since, in this case, the intention was to find gradually densifying subgraphs, the sampling method was based on adding all neighbors of nodes along a random walk. For more complex behavior patterns, however, other sampling methods will be necessary.

The application of the algorithms to the web proxy data utilized the presence of both sparse truth data and domain expertise. The sparse truth data described four cyber events that were used to both tune the algorithms and demonstrate their effectiveness. Domain expertise was used after the identification of other statistical anomalies to distinguish which anomalies corresponded to previously unknown truth events. In future efforts, collaboration with domain experts will be vital in order to evaluate the cues identified by the uncued statistical techniques.

A majority of the algorithms developed under the VLG study have linear computational complexity (commonly in number of relationships or edges). Even with efficient algorithms, however, computational limitations were commonly encountered. As previously mentioned, a parallel expressive associative array API will be necessary to support increasing problem scales. However, those will only improve capability (i.e., the ability to process large datasets) and not efficiency. Observed efficiency for graph and sparse array computations was often on the order of 10^{-3}–10^{-5}. This number is only expected to get worse as more complex data structures are analyzed (increased number of attributes, probabilistic graphs, etc.). To address this, full system designs (coupled across hardware, software, algorithms, and data) will have to be considered.

This page intentionally left blank.

8. SUMMARY

This report documents novel algorithmic developments in the analysis of massive dynamic graphs, all informed by real data. One data source is Thompson Reuters' Web of Science database, which is used to build citation and coauthorship graphs. The other source is web proxy logs, which are used to build a bipartite graph of computers connecting to web servers. Based on phenomena observed in the data, three new models for graph behavior were developed: a preferential attachment mechanism with memory, a weighted average of the previous connections in a stream of graphs, and a generalized linear model for modeling based on attributes. These algorithms have been demonstrated on the datasets of interest, and have proven, both in simulation and in real networks, to enhance detection performance when incorporated into the model for graph residuals. Empirical results include the detection of a botnet (after 15 minutes of activity) in the web proxy log, which, using current techniques, took 10 days to detect.

An ongoing follow-on study aims to expand this work by incorporating uncertainty into the graph analytics. This will allow the fusion of multiple observations, each with different trust levels, and to quantify the impact that this has on subgraph detection. Other avenues of investigation include continuing to scale to larger graphs, and documenting properties of graphs in a variety of application domains, allowing for algorithm development that spans the space of graphs in many dimensions.

This page intentionally left blank.

REFERENCES

[1] B. A. Miller, N. Arcolano, M. S. Beard, J. Kepner, M. C. Schmidt, N. T. Bliss, and P. J. Wolfe, "A scalable signal processing architecture for massive graph analysis," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5329–5332, 2012.

[2] N. T. Bliss and B. A. Miller, "Emergent behavior detection in massive graphs," *SIAM Conf. Parallel Process. for Scientific Computing*, 2012. Minisymposium 'Parallel Analysis of Massive Social Networks'.

[3] B. A. Miller and N. T. Bliss, "A stochastic system for large network growth," *IEEE Signal Process. Lett.*, vol. 19, no. 6, pp. 356–359, 2012.

[4] M. C. Schmidt, "Detection of anomalous events in large-scale graphs," in *Proc. Graph Exploitation Symp.*, 2012.

[5] N. Arcolano and B. A. Miller, "Statistical models and methods for anomaly detection in large graphs," *SIAM Ann. Meeting*, 2012. Minisymposium 'Massive Graphs: Big Compute Meets Big Data'.

[6] E. M. Rutledge, B. A. Miller, and M. S. Beard, "Benchmarking parallel eigen decomposition for residuals analysis of very large graphs," in *Proc. IEEE High Performance Extreme Computing Conf.*, 2012. To appear.

[7] T. Idé and H. Kashima, "Eigenspace-based anomaly detection in computer systems," in *Proc. KDD '04*, pp. 440–449, 2004.

[8] T. Mifflin, "Detection theory on random graphs," in *Proc. Int. Conf. Information Fusion*, pp. 954–959, 2009.

[9] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E*, vol. 74, no. 3, 2006.

[10] B. A. Miller, N. T. Bliss, and P. J. Wolfe, "Toward signal processing theory for graphs and non-Euclidean data," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5414–5417, 2010.

[11] B. A. Miller, N. T. Bliss, and P. J. Wolfe, "Subgraph detection using eigenvector L1 norms," in *Advances in Neural Inform. Process. Syst. 23* (J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds.), pp. 1633–1641, 2010.

[12] I. M. Johnstone and A. Y. Lu, "Sparse principal components analysis." arXiv:0901.4392v1, 2009.

[13] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[14] N. Singh, B. A. Miller, N. T. Bliss, and P. J. Wolfe, "Anomalous subgraph detection via sparse principal component analysis," in *Proc. IEEE Statistical Signal Process. Workshop*, pp. 485–488, 2011.

[15] B. A. Miller, M. S. Beard, and N. T. Bliss, "Matched filtering for subgraph detection in dynamic networks," in *Proc. IEEE Statistical Signal Process. Workshop*, pp. 509–512, 2011.

[16] J. Kepner et al., "Dynamic distributed dimensional data model (D4M) database and computation system," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5349–5352, 2012.

[17] K. H. Rosen, *Discrete Mathematics and Its Applications*. McGraw-Hill, fourth ed., 1999.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
20 August 2013	Project Report	

4. TITLE AND SUBTITLE

Very Large Graphs for Information Extraction (VLG)
Summary of First-Year Proof-of-Concept Study

5a. CONTRACT NUMBER
FA8721-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Benjamin Miller, Nadya Bliss, Nicholas Arcolano, Michelle Beard, Jeremy Kepner, Matthew Schmidt, and Edward Rutledge

5d. PROJECT NUMBER
2140

5e. TASK NUMBER
2

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

VLG-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intelligence Advanced Research Projects Activity
Office of Incisive Analysis
Washington, DC 20511

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In numerous application domains relevant to the Department of Defense and the Intelligence Community, data of interest take the form of entities and the relationships between them, and these data are commonly represented as graphs. Under the Very Large Graphs for Information Extraction effort—a one-year proof-of-concept study—MIT LL developed novel techniques for anomalous subgraph detection, building on tools in the signal processing research literature. This report documents the technical results of this effort. Two datasets—a snapshot of Thompson Reuters' Web of Science database and a stream of web proxy logs—were parsed, and graphs were constructed from the raw data. From the phenomena in these datasets, several algorithms were developed to model the dynamic graph behavior, including a preferential attachment mechanism with memory, a streaming filter to model a graph as a weighted average of its past connections, and a generalized linear model for graphs where connection probabilities are determined by additional side information or metadata. A set of metrics was also constructed to facilitate comparison of techniques. The study culminated in a demonstration of the algorithms on the datasets of interest, in addition to simulated data. Performance in terms of detection, estimation, and computational burden was measured according to the metrics. Among the highlights of this demonstration were the detection of emerging coauthor clusters in the Web of Science data, detection of botnet activity in the web proxy data after 15 minutes (which took 10 days to detect using state-of-the-practice techniques), and demonstration of the core algorithm on a simulated 1-billion-vertex graph using a commodity computing cluster.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	Same as report	56	